The image size is the number of words in the data field. The time stamp is 32-bit UNIX time followed by a 16-bit sequence number.

The dsp code includes a host port communications controller. The host port of the dsp 17 is a 24-bit to 8-bit asynchronous parallel interface. It appears mapped in the I/O space of the processor in the communications block 10 as a collection of 8-bit registers. There are three transmit and receive registers. These represent the high, middle and low bytes of a single dsp word.

The processor in the communications block 10 sends "host commands" to the dsp 17 by writing 8-bit values to a register. These commands trigger a vectored interrupt in the dsp 17. There are four services the microprocessor in the communications block 10 can request from the dsp, these are:-

- Request a Local BUS packet from the dsp.
- Request and internal packet from the dsp.
- Request the dsp to get ready to receive a Local BUS packet.
- Request the dsp to get ready to receive an internal packet.

The host port communications controller software module on the dsp uses the dsp's Direct Memory Access controller to send the data to the host port. This saves on core processing and BUS usage.

The DVC maintains a real time clock which is used to add a time stamp to the images as they are stored. This clock can be updated from the Local BUS.

The dsp code includes message buffering. In addition to sending out images over the host port the DVC also sends and receives other Local BUS and internal packets. The message buffering simply queues up the messages to be sent or the messages received until they can be sent or processed respectively.

Finally, the dsp includes as a major component supervisory routines. The DVC runs several supervisory routines.

There is a check for the integrity of the code, a check for the correct functioning of memory and three watchdog routines. The watchdog routines check for hanging of the image processing, the sensor communications and the frame capture. If any problems are detected they are reported to the Local BUS either directly if the dsp 17 is capable, or indirectly by a termination mode via the controller of the communications block 10.

Major parts of the code of the controller of the communications block 10 are a dsp controller, Local BUS port controller and supervisory routines.

The dsp controller controls all of the interfacing to the dsp17. It controls such things as the down-loading of code to the dsp, the hang supervision (watchdog) and the sending and receiving of packets from and to the dsp (including images).

The dsp code is downloaded to the dsp whenever the dsp is reset. The bootstrap code on the dsp waits for the length and address of the code to be sent to the host port and then captures the code.

The Local BUS port controller is a serial communications controller peripheral (SCC) built into the same chip as the microprocessor of the communications block

10. The SCC handles much of the low level detail of the Local BUS communications. The Local BUS controller code handles the higher level part of the Local BUS protocol and processes the messages destined for the communication block 10 microprocessor.

The dsp has no hardware watchdog for hang prevention. Instead, the microprocessor of the communications block 10 functions as a watchdog. The dsp must send interrupts to the microprocessor periodically. If the microprocessor goes for a certain time without receiving an interrupt from the dsp it assumes the dsp has hung. It then checks the host port for a termination code from the dsp and this is used to help determine what has gone wrong with the dsp.

The communications block microprocessor supervisory routines also run some self-monitoring of the code integrity and memory functionality.

Aspects of the present invention have been described by way of example only and it should be appreciated that modifications and additions may be made thereto without departing from examples from the scope of the appended claims.